

# Bootcamp Full Stack

## Proyecto Integrador: Etapa 2

### Consideraciones generales

Para la segunda etapa del proyecto se deben seguir las mismas especificaciones que en la primera. Se permiten cambios en los estilos, agregado de elementos y rediseño completo siempre y cuando se cumplan con los requisitos para ambas etapas.

En cuanto al funcionamiento del sitio, se debe crear una SPA que permita acceder a cada vista sin recargar el header ni el footer. La comunicación con el servidor puede intercambiar cualquier tipo de datos y se pueden obtener de forma asincrónica en cualquier momento.

Se puede utilizar HTML, CSS y Javascript en su forma avanzada ó ReactJS en un diseño basado en componentes. El entorno elegido se deberá conectar con un Backend implementado en una cuenta de mockapi.io, y se deberá usar fetch() ó Axios según corresponda con async/await para las conexiones asincrónicas con los servidores para obtener recursos locales (live server para diseños en HTML / CSS y Javascript) ó remotos (mockapi.io). El historial del navegador registrará la navegación interna de la aplicación frontend.

Como opción, se podrán brindar feedbacks para todas las acciones que realice el usuario o mensajes emergentes para que el usuario comprenda lo que está sucediendo. Se debe tener en cuenta las distintas resoluciones de pantalla y tipos de dispositivo, y acomodar los elementos apropiadamente en todas ellas.

### Descripción de lo que se debe desarrollar en esta segunda etapa

El sitio web a desarrollar tendrá un botón de carrito en el header que mostrará un modal con el listado de productos que se hayan agregado al carrito. El modal se cerrará al presionar nuevamente en el botón del carrito, la x de cierre incorporada dentro del modal, la tecla ESC del teclado o al hacer clic en cualquier zona fuera del modal. Otra opción de implementación del carrito podrá ser que este tenga un link en el menú de navegación y su contenido

aparezca como una vista más. Dentro del modal ó la vista del carrito se mostrarán los productos que hayan sido agregados al carrito a partir de las cards. Cada producto agregado al modal/vista mostrará su foto, nombre, precio, cantidad y subtotal. Adicionalmente, se podrán agregar botones para eliminar el ítem individual ó total y para sumar o restar la cantidad o editar directamente el número de unidades.

Un vez que confirmamos los items, podremos enviar el pedido que se va a guardar en un recurso creado en mockapi.io llamado **carrito**

Todas las vistas deberán recibirse de manera asíncrona y la home mostrará el listado de cards generadas de forma programática a través de Javascript ó ReactJS, que permitan agregar el producto que se detalla en ella al carrito. Al hacerlo, se deberá informar al usuario con un mensaje que se agregó el producto al carrito y se adicionará el producto al modal / vista del carrito como uno nuevo, si no existía previamente.

En la vista de alta, se deberá validar el nombre, precio, stock, marca y categoría del producto. Además, cada campo deberá informar los errores de validación al perder el foco y si se intenta enviar el formulario y al menos un campo no cumple con los requisitos, se mostrarán los errores correspondientes y el formulario no se enviará. Solo se permitirá el envío de los datos si todos los campos fueron validados exitosamente.

Como opción se podrá incorporar una tabla debajo del formulario de ingreso, que permita ver el detalle de productos dados de alta y poder modificarlos y eliminarlos en forma individual.

## Sección "Home"

Se mostrará un listado de cards generadas de forma programática en HTML a partir de un array definido con JS ó de componentes de ReactJS precargados (cualquiera de los dos) con la información de los productos que se encuentran en mockapi.io, recurso **productos**.

Cada card tendrá un botón para agregar el producto al carrito y se mostrará un mensaje no invasivo al usuario cuando se agregue.

Al agregarse un producto al carrito, se adicionará al modal/vista del carrito como uno nuevo y si ya existía, se incrementará la cantidad de unidades sin repetir el producto.

Se pueden agregar elementos interactivos a las cards de forma optativa, como descripciones largas del producto o fotos adicionales. Se registrarán los eventos que correspondan en cada card para que está reaccione con las acciones esperadas.

## **Sección "Alta"**

El formulario definido deberá contar con las validaciones detalladas para los siguientes campos: nombre, precio, stock, marca, categoría, descripción corta, descripción larga, envío sin cargo, edad desde, edad hasta, y foto.

Cada campo deberá informar si tiene errores de validación al perder el foco y al intentar enviar el formulario. Si al menos un campo no cumple con los requisitos, el formulario no deberá enviarse y cada campo deberá mostrar los errores correspondientes.

Solo se permitirá el envío de los datos si todos los campos fueron validados exitosamente. Se mostrará al usuario qué tipo de validaciones se están realizando para que sepa qué correcciones debe realizar al completar el formulario.

Se pueden agregar otros campos al formulario si se desea.

Si los datos son consistentes, se enviarán a mockapi.io en un recurso generado llamado **productos**

## **Sección "Contacto"**

El formulario de contacto deberá tener el mismo comportamiento que el formulario de Alta y se podrán agregar otros campos al formulario si se desea. Los campos requeridos son: nombre, email y comentarios.

***El plazo de entrega máximo es de 20 días.***